



2022-09-15

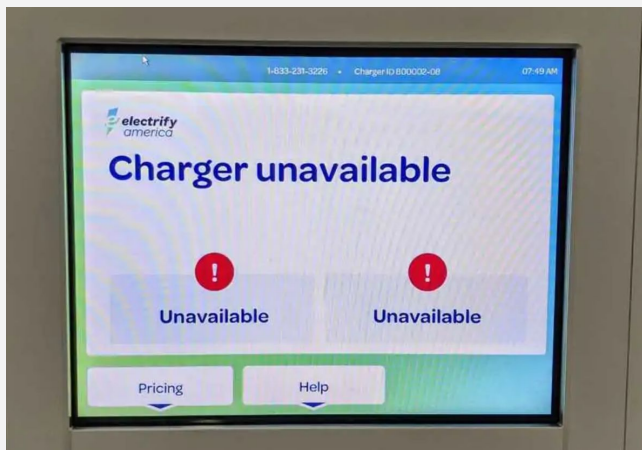
EVERest @ Embedded Linux Conference 2022

Technical Introduction to EVERest:

Open Source Firmware for EV Charging Stations

How to charge a car?

Charging a car is easy, RIGHT?

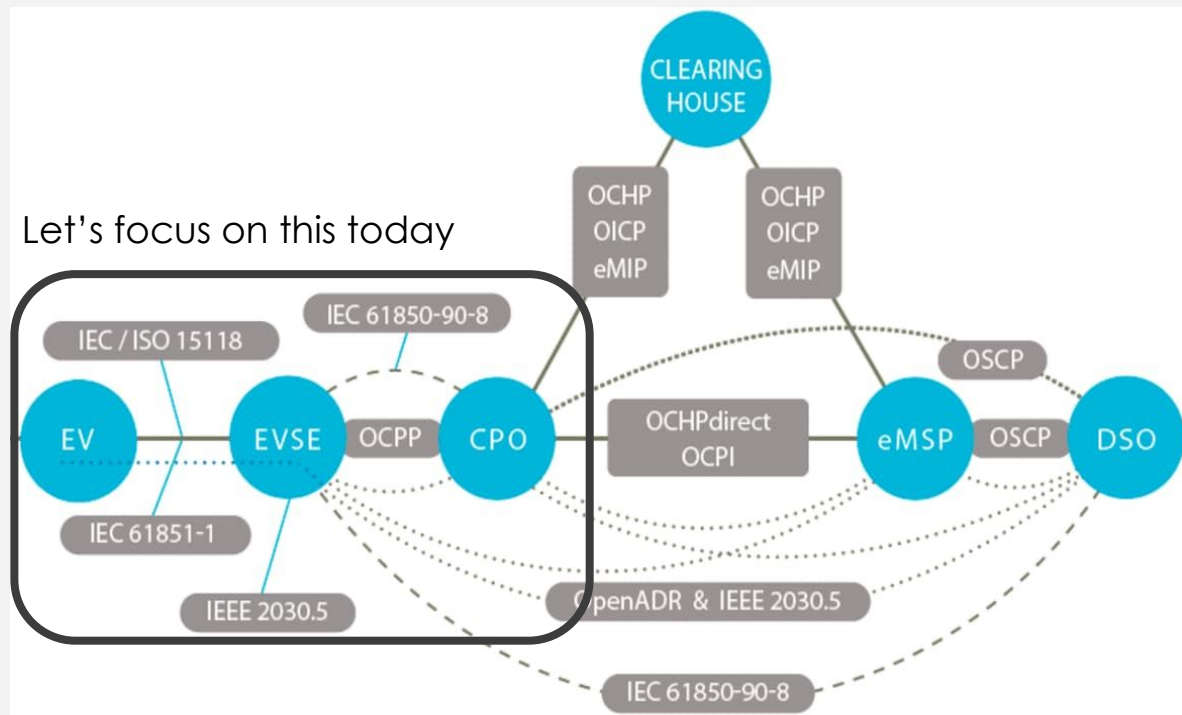


<https://insideevs.com/news/388501/electrify-america-discusses-network-problems/>



That is how “easy” charging really is

Let's focus on this today



<https://www.emobilitysimplified.com/2020/08/ev-roaming-protocol-differences-OCPI-OICP-OCHP-eMIP.html>

- **EV**
Electric Vehicle
- **EVSE**
Electric Vehicle Supply Equipment
- **CPO**
Charge Point Operator
- **Clearing House**
- **eMSP**
eMobility Service Provider
- **DSO**
Distribution System Operator (Utility)

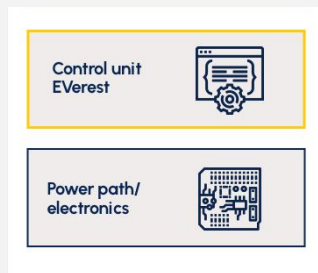
Connecting car & charger

Protocols

- A) IEC 61851
- B) DIN SPEC 70121
- C) ISO 15118 (multiple versions)
- D) CHAdeMO

PHY

- A) CP/PP/1kHz PWM/V
- B) CAN BUS
- C) PLC (GreenPhy)



Power

- A) AC 110/230V, 1/3 phases, 6-63A
- B) DC 400/800V, 0-1000A
- C) Inductive
- ⇒ each: Unidirectional / Bidirectional
- ⇒ Metering: V/A/f, German metering law, ..
- ⇒ Electrical Safety (RCD)

Connector designs listed in IEC 62196-2 and -3

Power Supply	United States	European Union	Japan	China
1-phase AC (62196.2)	 Type 1 (SAE J1772)	 Type 2 (DE, UK)	 Type 1 (SAE J1772)	 Type 2 (GB/T 2034.2)
3-phase AC (62196.2)	 Type 2 (SAE J3068)	 Type 3 (IT, FR; now deprecated)	N/A	
DC (62196.3)	 EE (CCS Combo 1)	 FF (CCS Combo 2)	 AA (CHAdeMO)	 BB (GB/T 2034.3)
			 ChaoJi (planned)	

EU



https://en.wikipedia.org/wiki/Charging_station

Connecting charger to cloud



OCPI, OCHP,

**Roaming +
Different Networks**



>> 150 dialects

- A) OCPP 1.2 / 1.6 / 2.0.1
- B) IEC63110
- C) MQTT & OTHERS ;-)

Protocols

XML / JSON

Web Sockets

VPN / Sim+private APN
/ HTTPS



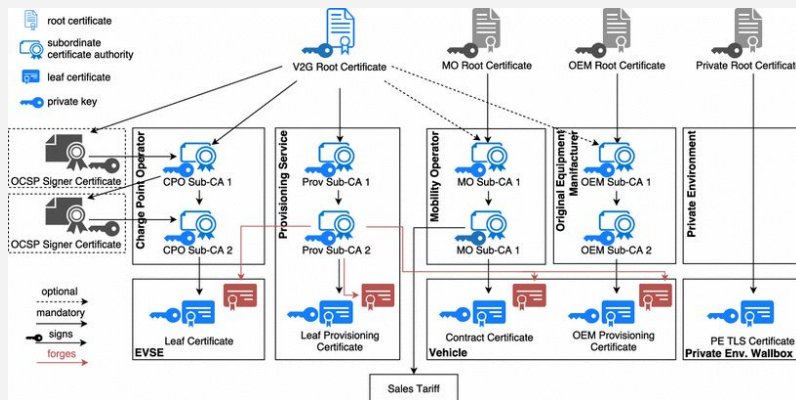
Control unit
Everest



Power path/
electronics

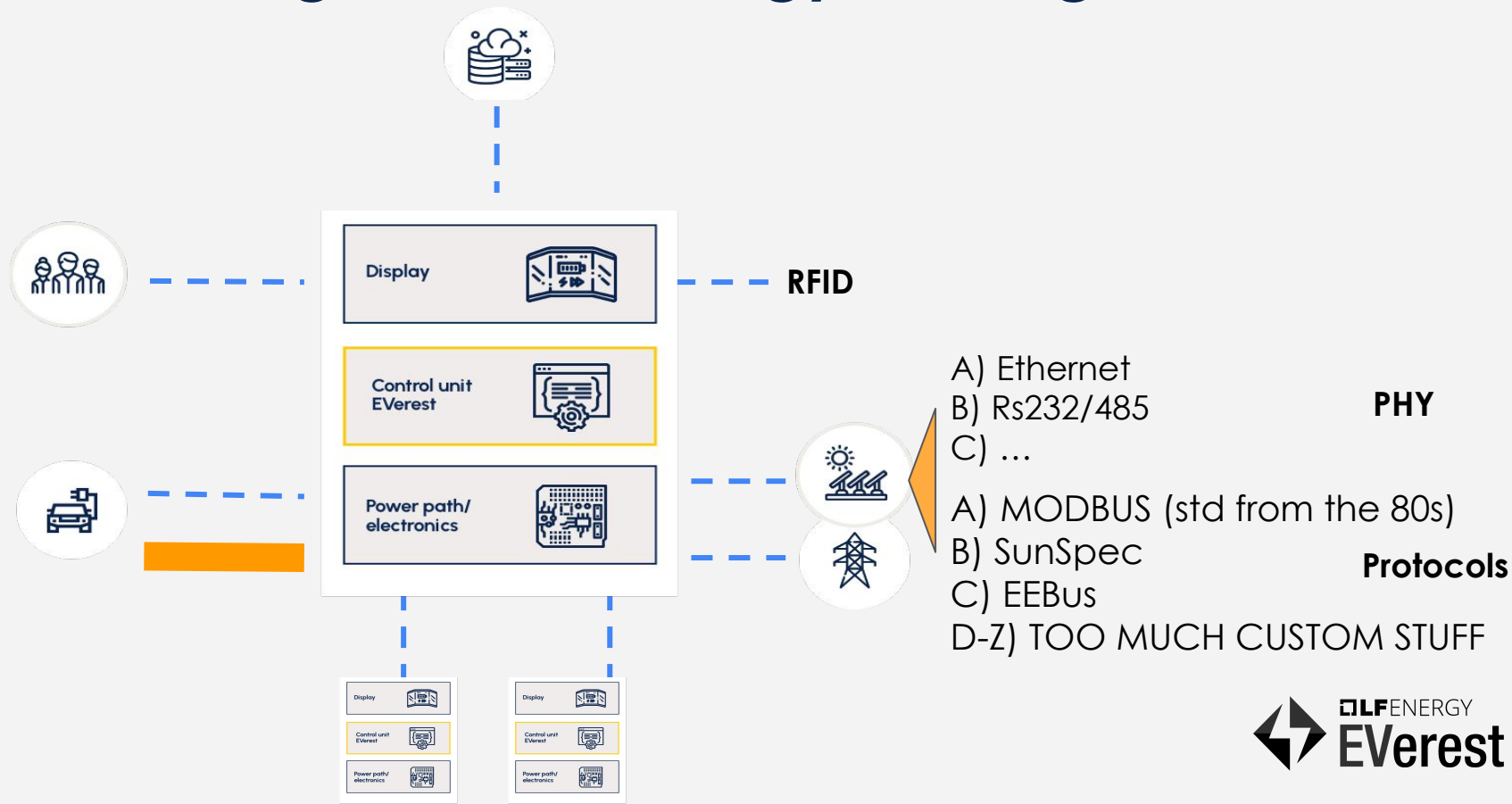


Plug & Charge will be here in 2 years™
original since 2014
Rollout of complex distributed PKI

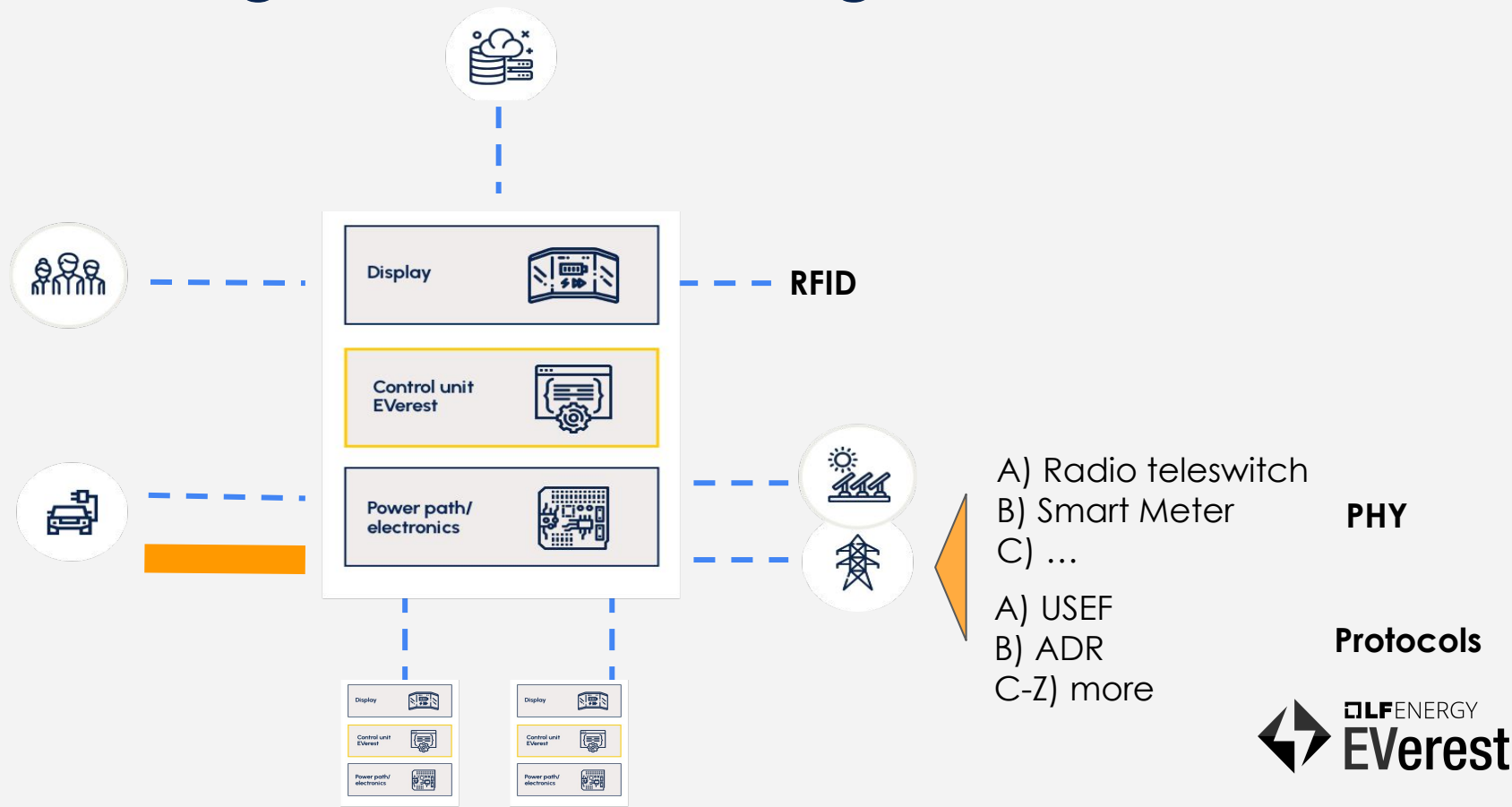


<https://link.springer.com/article/10.1007/s00450-017-0342-y>

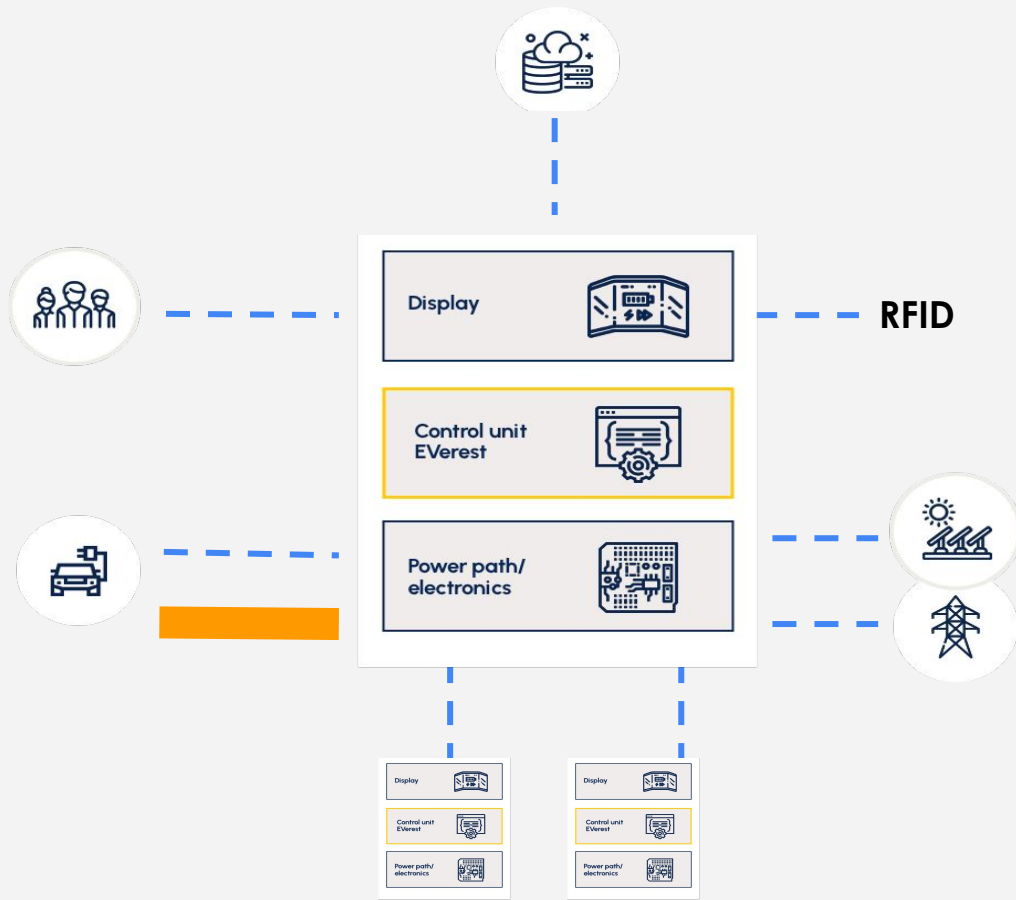
Connecting to local energy management



Balancing with the electric grid



It's a bit of a mess! And it is even growing



No De-facto SW stack!

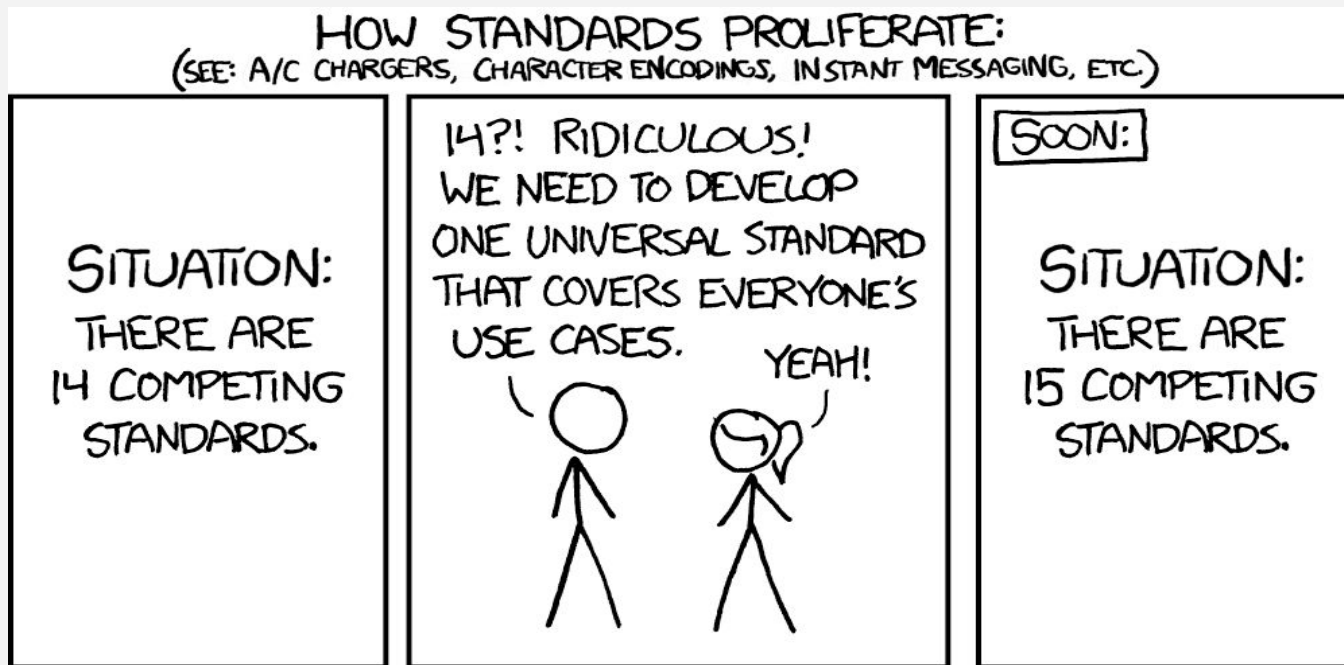
- Too many links
- Too many standards
- Too many Dialects
- Ever growing



- High fault-rate, poor UX
- Expensive & slow development
- Complex mechanisms to proliferate innovations
- Market fragmentation

Customers and industry suffering

We cannot fix this by adding more standards...

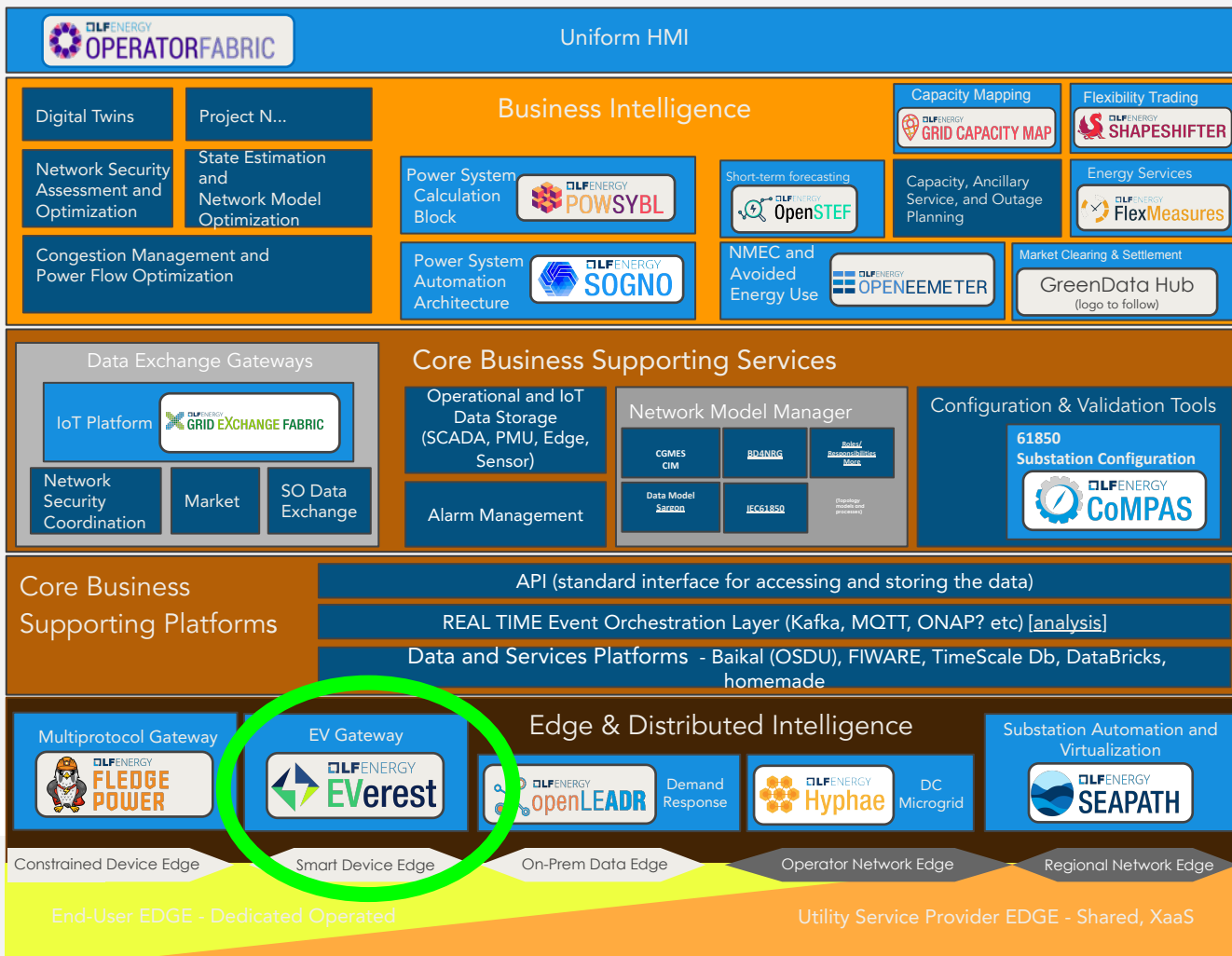


Everest provides a standard compliant open source implementation that serves as a defacto standard.

Application Catalogue

Data & Services

Infrastructure

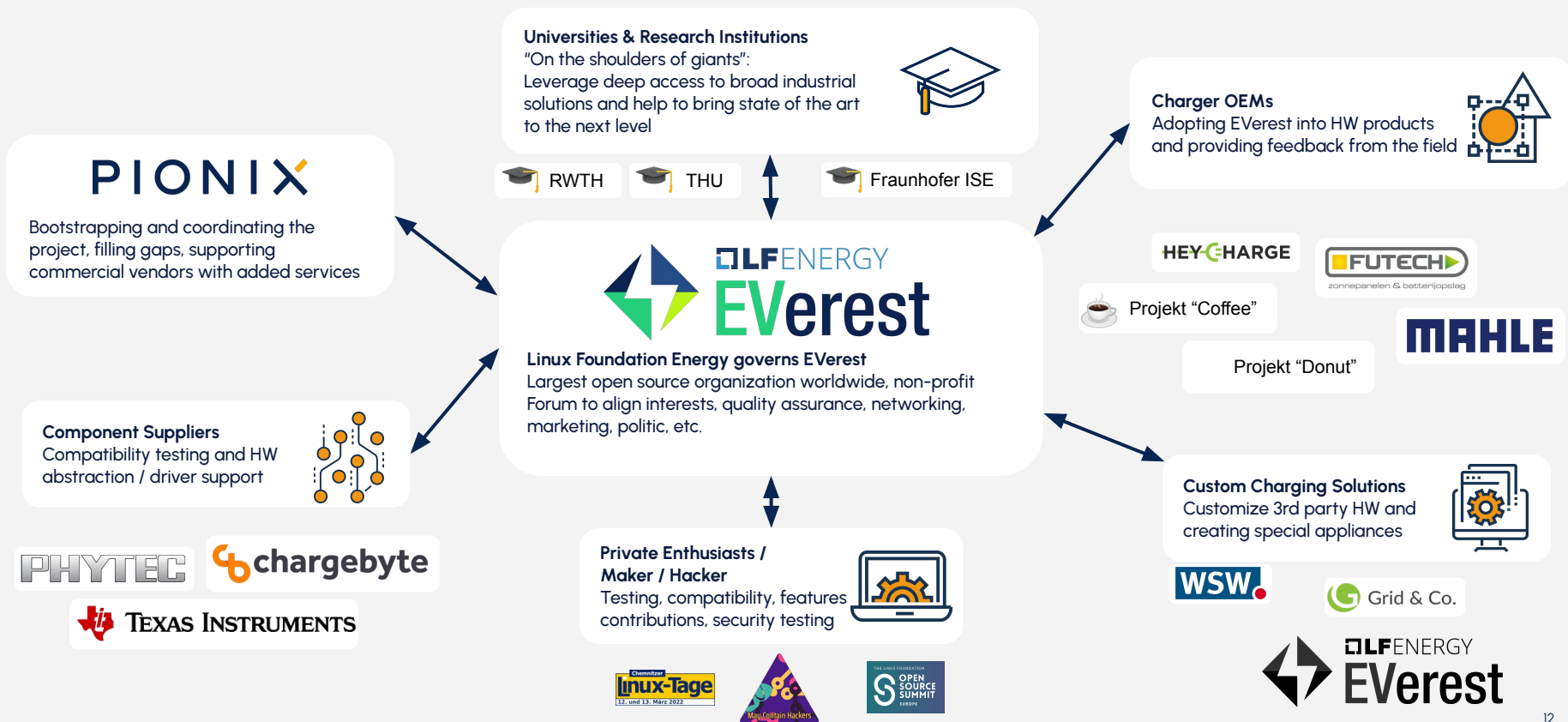


LF Energy
Current
Projects

Future
LF Energy
Projects

Security

EVERest is an open source charger SW for a broad community



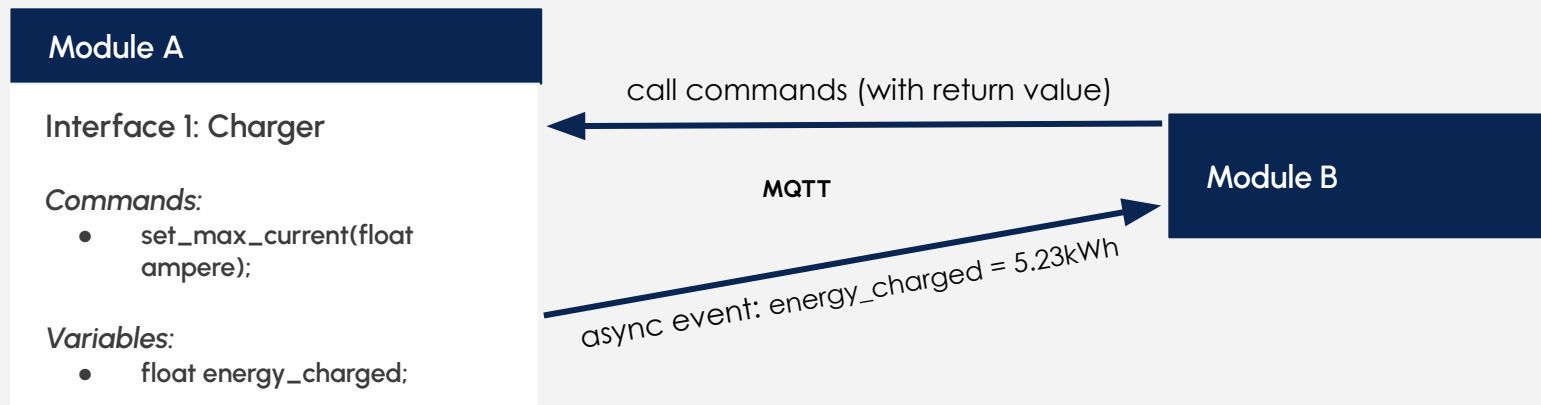
What is EVerest?

EVERest is a complete operating system for EV chargers

- Provides all functionality needed from smart home chargers to commercial fast chargers
- Runs on tiny embedded linux
- Aims to support as many different HW platforms as possible
- Commercial friendly open source (Apache 2.0 licensed)
- Don't reinvent the wheel: Use EVERest as common base layer and add some unique features of your product on top!

<https://github.com/EVERest>

EVerest framework: Modular architecture

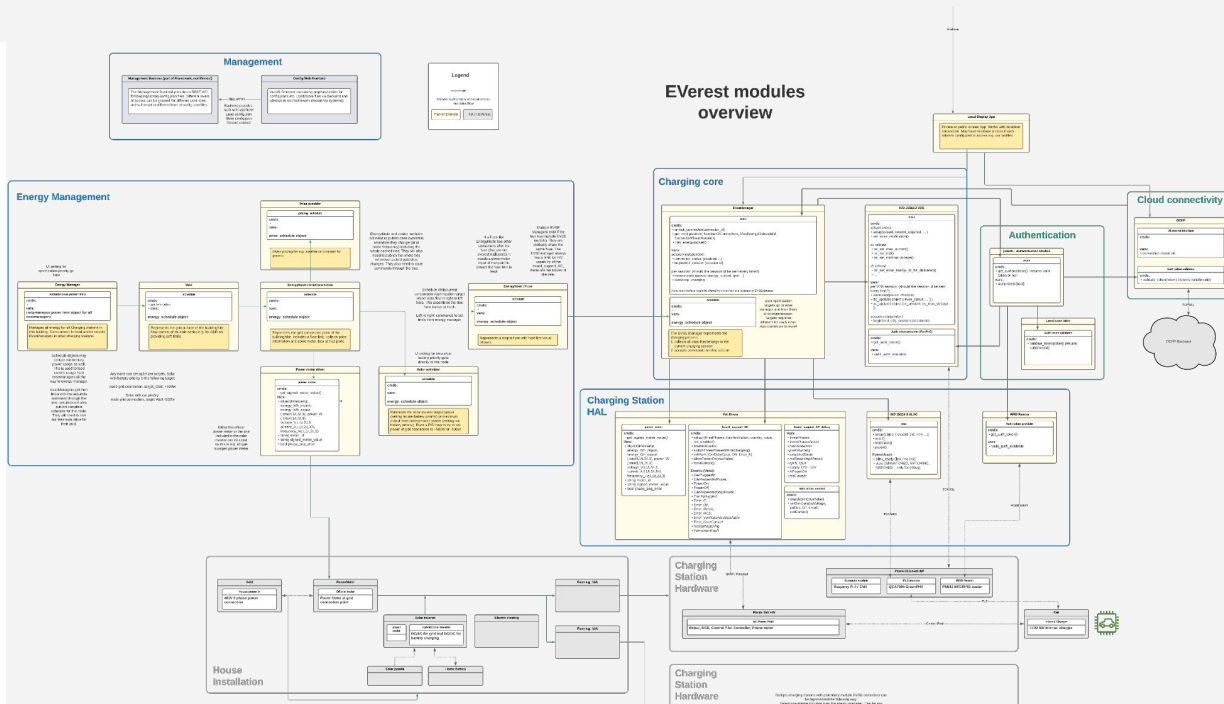


- Modules are individual Linux processes, can be loaded multiple times
- Modules expose (multiple) interfaces on MQTT
- Config file to connect module requirements/interfaces
- Can run on different computers / can spread over multiple chargers!
- Framework starts/stops/restarts modules

- Communication via MQTT broker, transparent to modules
- Loose coupling: A does not know B, B does not know A
- Framework manages synchronous RPC, argument type safety, dependency checking
- Additional services: Logging, configuration management, code auto generation, ...

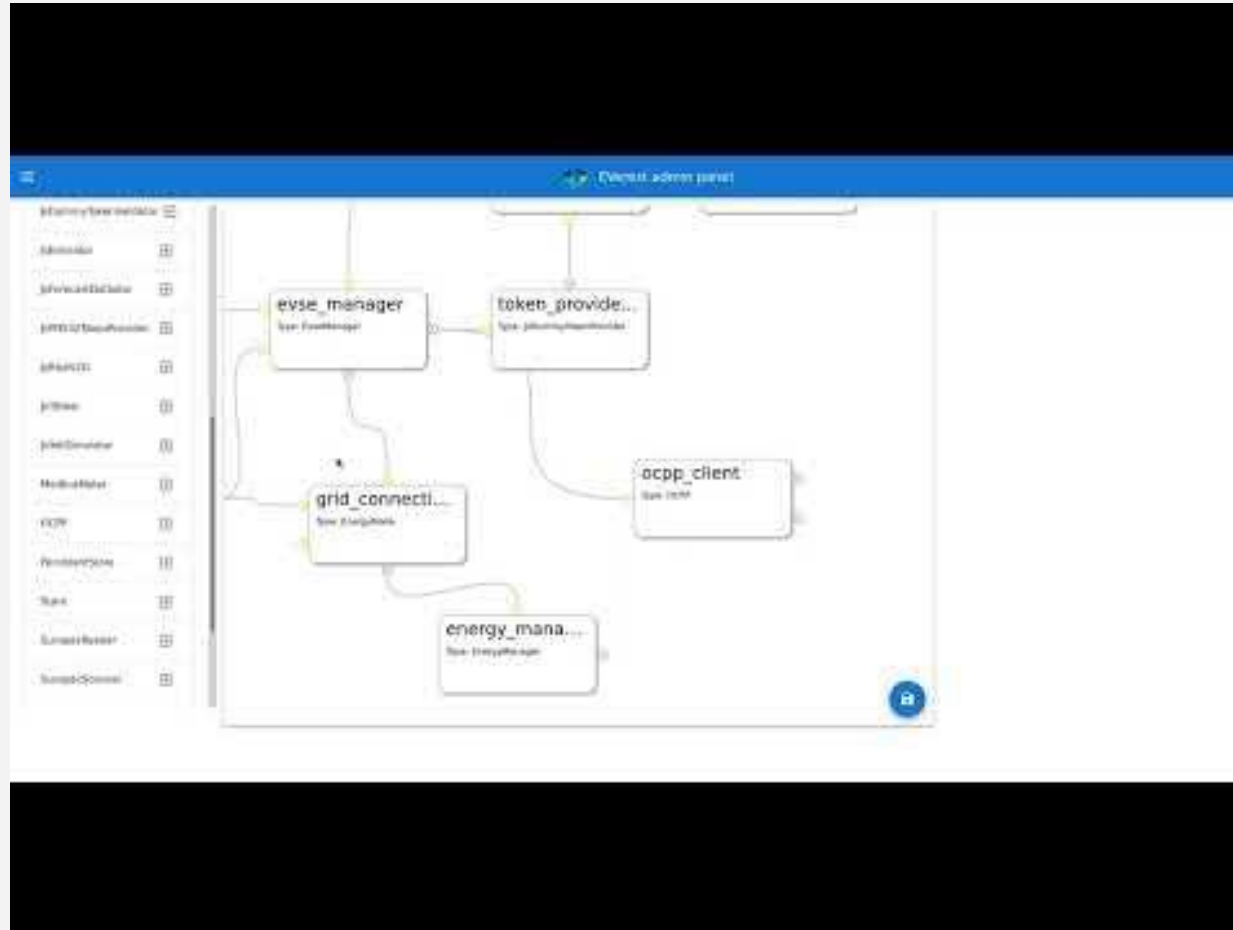
Module config file

- JSON config file which modules to load/connections represent HW
- E.g. if your wallbox does not support RFID, simply do not load the module
- If it has 2 charging connectors, load the charging core modules twice
- in development: web interface to graphically edit module connections



https://lucid.app/lucidchart/5364a561-1881-414a-a33a-7ba29425a140/edit?page=0_0#

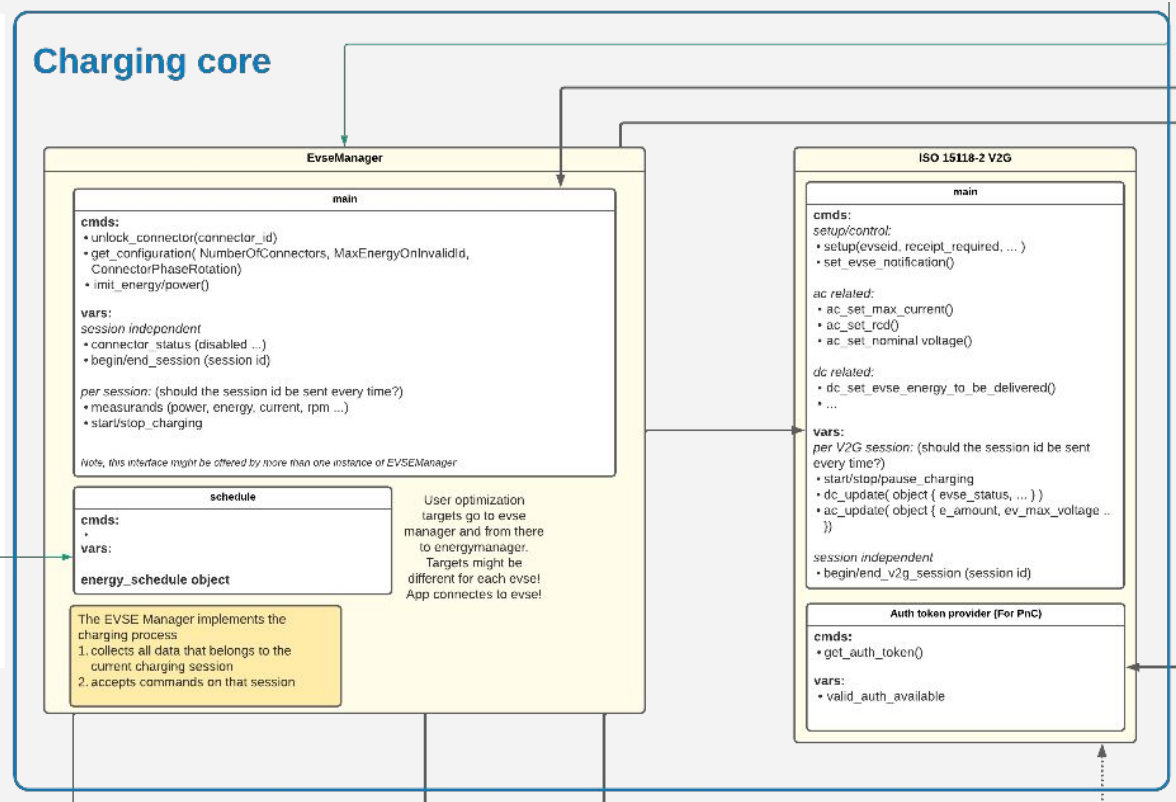
Graphical Setup Webinterface



<https://youtu.be/xZQtN8vXMlk>

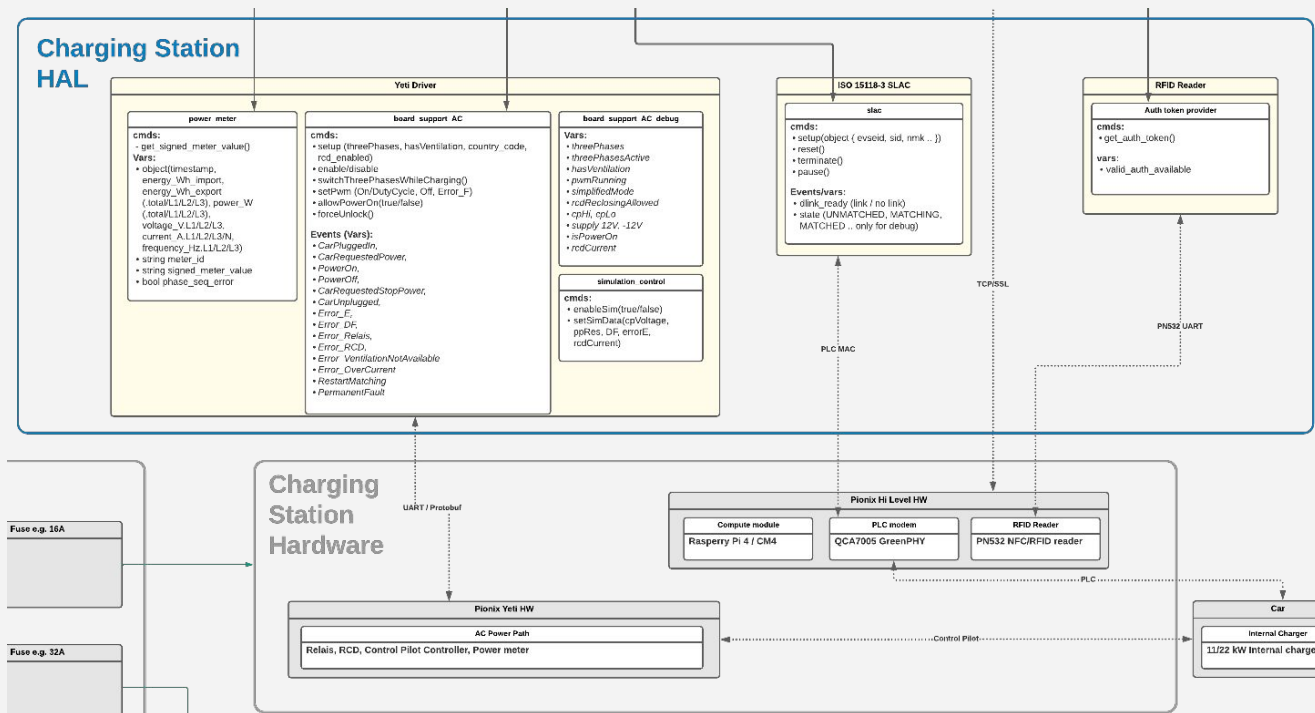
Charging core

- EVSE: central charging module that controls CP signal and manages IEC61851 and ISO15118 interaction, charging session, ...
- ISO15118 protocol module
- ISO15118 support is under development with RISE-V2G, aiming for a complete rewrite including ISO15118-20 and DIN SPEC



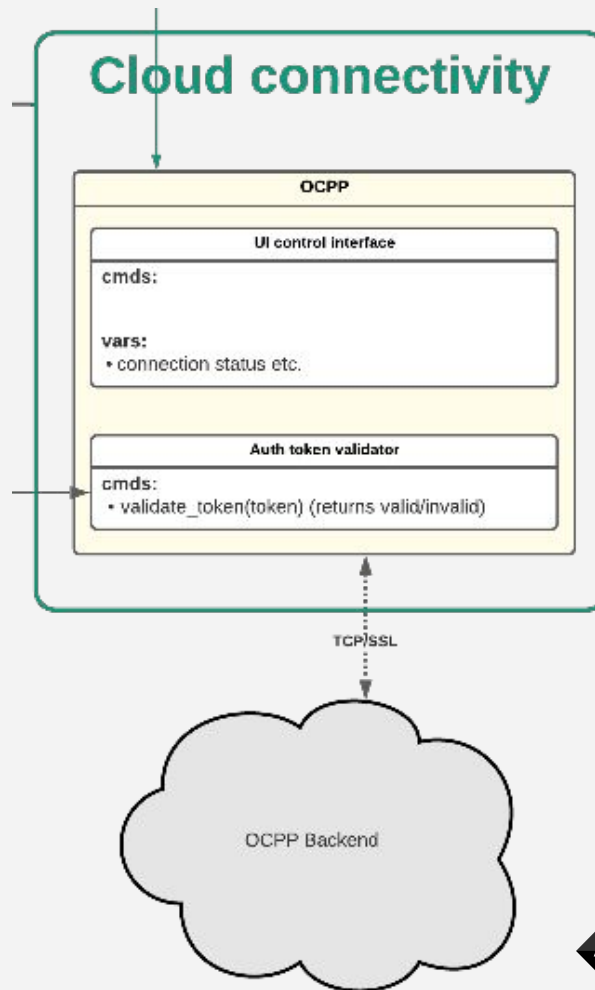
Charging Station Hardware Abstraction Layer

- Simple interface to be easily portable
- Supports Pionix dev kit for now
- Porting to new hardware requires drivers for:
- CP signal PHY, Relais, RCD, Power meter, SLAC (if different from QCA7000), NFC/RFID reader

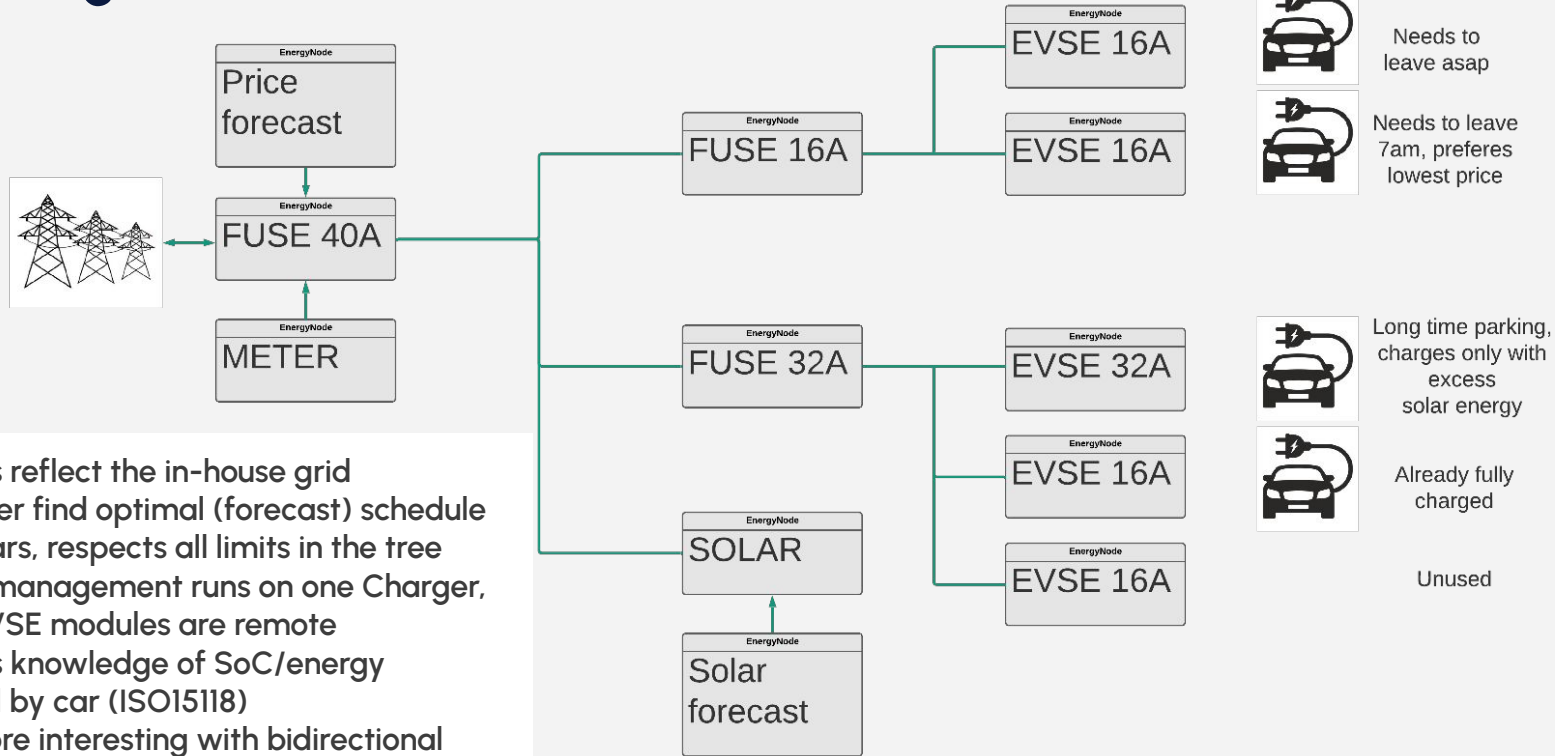


Backend connectivity

- Communication with the cloud for payment and management
- OCPP 1.6 JSON
- Try to make it standard compliant
- Verify with as many commercial backends as possible
- Support for optional profiles such as Smart Charging



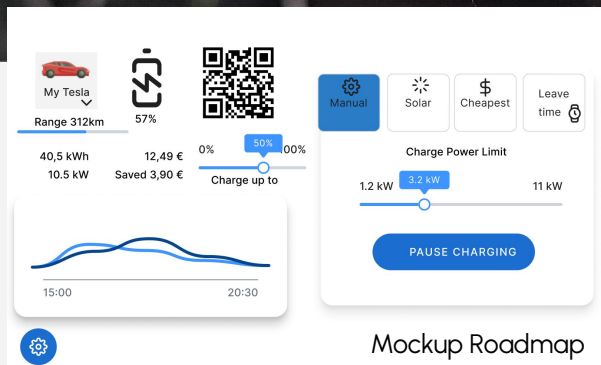
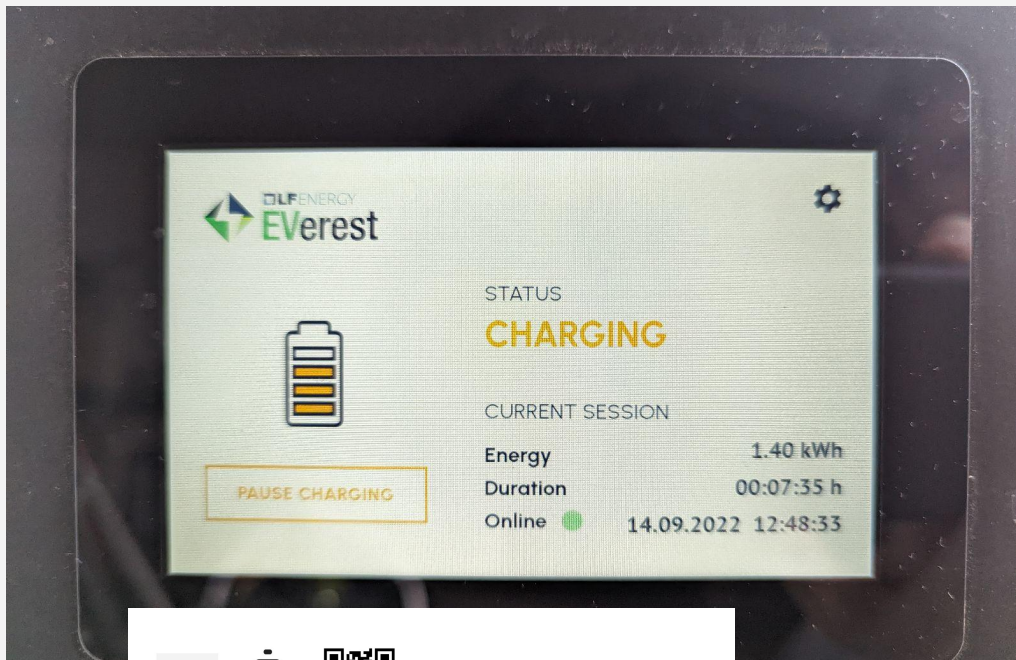
Energy Management



- Modules reflect the in-house grid
- Optimizer find optimal (forecast) schedule for all cars, respects all limits in the tree
- Energy management runs on one Charger, other EVSE modules are remote
- Requires knowledge of SoC/energy required by car (ISO15118)
- Gets more interesting with bidirectional support
- Under heavy development right now

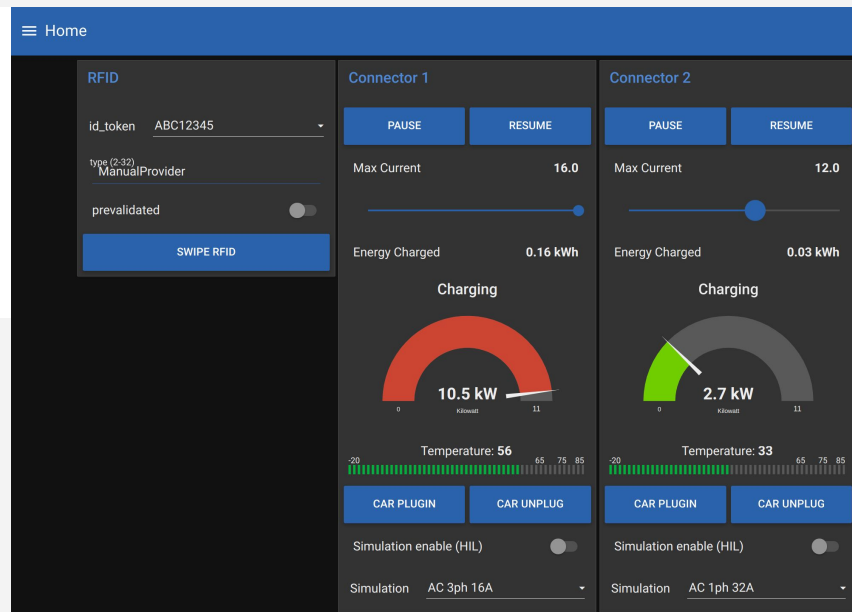
Local display app

- Based on flutter
- Runs locally on charger's display using flutter-pi
- Ongoing development



Software in the loop simulation

- Simulation for complete HAL
- Car simulator module that simulates everything a car can do on IEC61851
- You can run EVerest on your laptop and charge simulated cars with it
- ISO15118 SIL



Engineering toolset

- Build systems
- Build VM
- NodeRED ➡

Node-RED Dashboard

Home

EVSE Configuration

Simulation

Solar Manager Settings

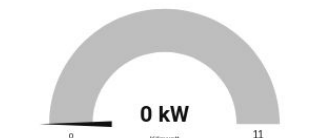
Debug

Charger

Maximum Current 16.00000

Energy Charged -0.00

Idle



Temperature: 37
-20 65 75 85

Control

MaxCurrent



PAUSE CHARGING

RESUME CHARGING

Switch optimizer

Select option

Price limit (EUR / kWh)

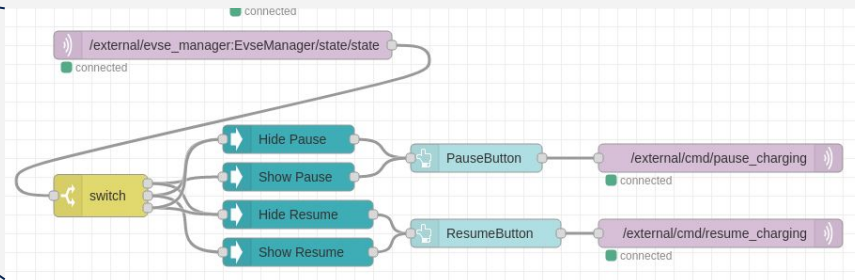
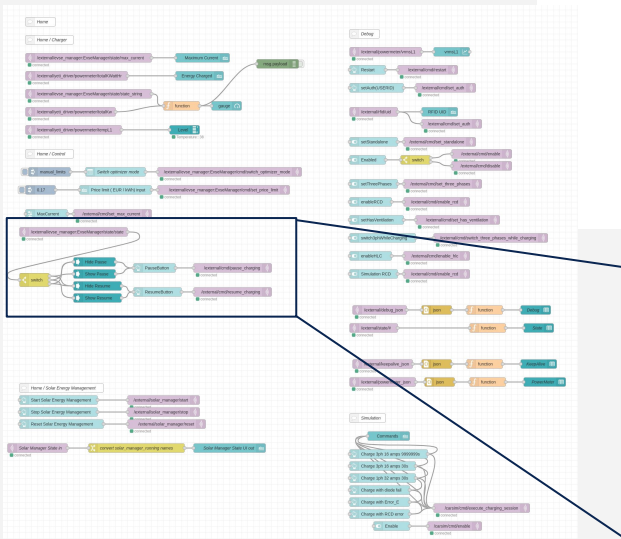
Solar Energy Management

START SOLAR ENERGY MANAGEMENT

STOP SOLAR ENERGY MANAGEMENT

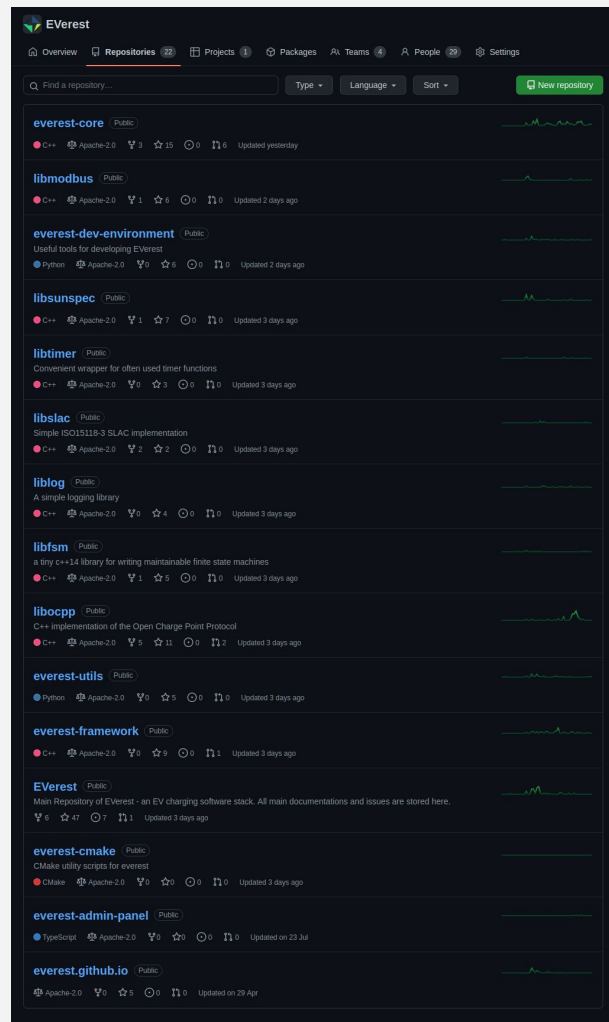
Solar Manager State

RESET SOLAR ENERGY MANAGEMENT



How to get involved

- GitHub <https://github.com/EVERest>
- **Weekly Dev Sync** - Tuesday's 10:00 CEST/CET
- Technical Steering Committee meeting for EVERest Monthly, every 4th thursday
Next: 22nd. September - 17:00 CEST
- Mailing list everest@lists.lfenergy.org
<https://lists.lfenergy.org/g/everest>
- Slack
https://join.slack.com/t/lfenergy/shared_invite/zt-f9z7c305-SYbhNb2ffXJ~8jGd_gpwWQ



Get in contact and contribute to EVerest



Kai

GitHub

<https://github.com/EVerest>

Join us along our journey to unify
the EV charging ecosystem with
open source.

On the Linux Foundation

[https://www.lfenergy.org/project
s/everest/](https://www.lfenergy.org/project-s/everest/)

Mailing list

everest@lists.lfenergy.org



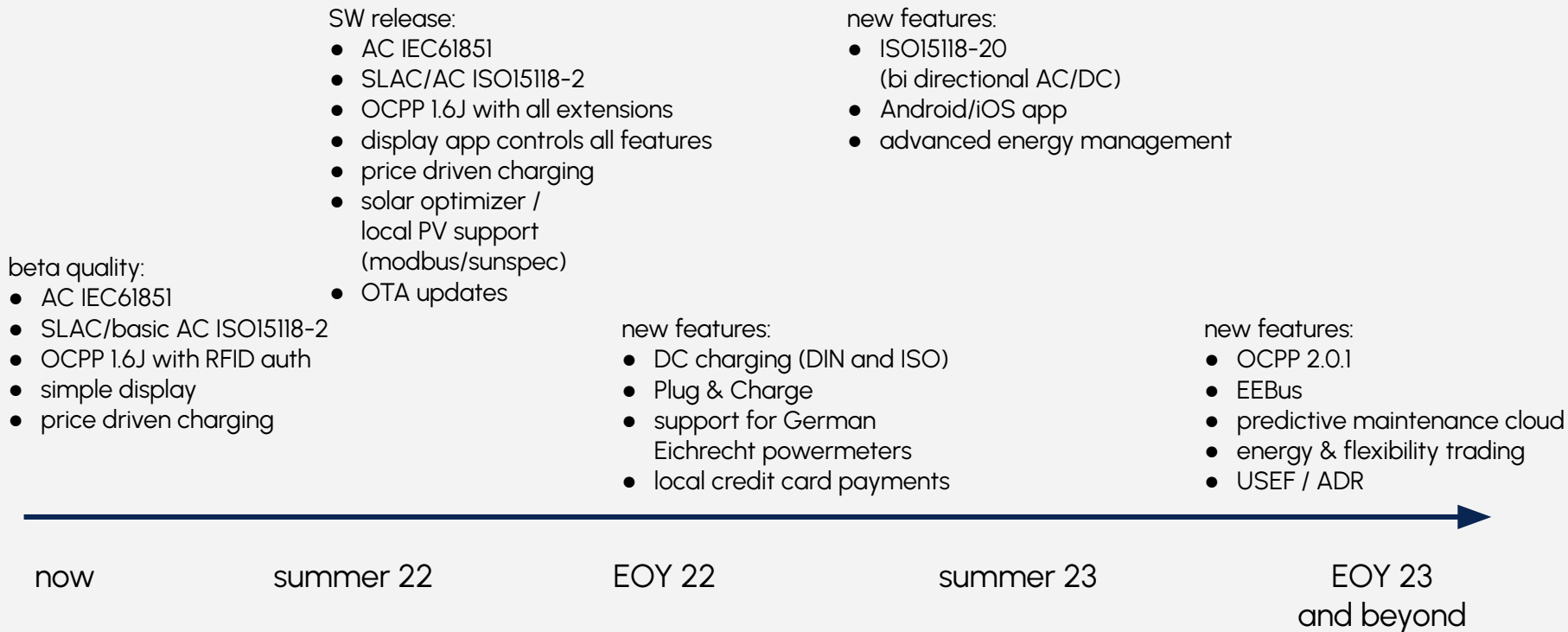
Piet

Slack

[https://join.slack.com/t/lfenergy/shared_invite/
zt-f9z7c305-SYbhNb2ffXJ~8jGd_gpWwQ](https://join.slack.com/t/lfenergy/shared_invite/zt-f9z7c305-SYbhNb2ffXJ~8jGd_gpWwQ)

Discussion

Everest Roadmap



...prioritization **pending community priorities...**